JOURNAL OF BASIC SCIENCE AND ENGINEERING

Privacy-Aware Hierarchical Federated Learning (PA-HFL): A Secure Multi-Tier Aggregation using Differential Privacy and Homomorphic Encryption

Mohammed Nizar Faruk[1], Sivaneasan Bala Krishnan[2]

[1]Department of Computer Science & Engineering,Navodaya Institute of Technology, Raichur, Karnataka, India

[2]Electrical Power Engineering Programme, Singapore Institute of Technology, Singapore.

DOI: https://doie.org/10.10399/JBSE.2026872499

Keywords:

Differential Privacy (DP)
Secure Multi-Party Computation (SMPC)
Homomorphic Encryption (HE)
Federated Learning (FL)
Hierarchical Federated Learning (HFL)
Blockchain

ABSTRACT

Privacy-preserving distributed learning has become critical for data-intensive and sensitive domains such as healthcare, smart cities, and industrial IoT. This paper presents a Privacy-Aware Hierarchical Federated Learning (PA-HFL) framework that integrates multi-tier aggregation with Differential Privacy (DP), Secure Multi-Party Computation (SMPC), and Homomorphic Encryption (HE) to address scalability, communication efficiency, and privacy leakage challenges inherent in conventional federated learning systems. The proposed architecture extends traditional two-tier federated learning into a hierarchical structure comprising local clients, edge/regional aggregators, and a global server, enabling decentralized yet coordinated model training while minimizing communication overhead. Privacy-enhancing technologies are strategically deployed across hierarchical layers: DP introduces calibrated noise to protect individual contributions, SMPC ensures secure intermediate computations without revealing private updates, and HE enables encrypted aggregation of model parameters throughout the learning lifecycle. Furthermore, a private blockchain is incorporated to ensure integrity, auditability, and tamper-resistant verification of model updates across all tiers. The framework is evaluated using extensive simulations on benchmark datasets including MNIST, Fashion-MNIST, CIFAR-10, and CIFAR-100, under both IID and non-IID data distributions. Experiments are conducted using Python, PyTorch, TenSEAL, and the Flower federated learning framework on a controlled hardware environment. Performance is assessed using metrics such as model accuracy, communication overhead, computational latency, scalability, and resistance to privacy attacks including reconstruction, membership inference, and property inference. Results demonstrate that the proposed PA-HFL framework achieves higher model accuracy and faster convergence compared to conventional federated averaging, while maintaining strong privacy guarantees.

Corresponding Author:

Mohammed Nizar Faruk
Department of Computer Science & Engineering,
Navodaya Institute of Technology (Autonomous), Raichur, Karnataka, India

1. INTRODUCTION

Hierarchical Federated Learning extends traditional federated learning by introducing multi-tier aggregation, typically involving local, regional, and global aggregation levels, which enhances scalability and reduces communication overhead while still aiming for high model accuracy [1]. This architecture inherently addresses the scalability challenges of two-tier FL systems, particularly when dealing with a large number of geographically distributed devices [2]. Furthermore, the integration of privacy-enhancing technologies like differential privacy, secure multi-party computation, and homomorphic encryption within this multi-tiered framework fortifies data confidentiality and model integrity against various adversarial threats [3], [4]. This advanced approach enables a more robust and efficient distributed machine learning paradigm, particularly crucial in scenarios with sensitive data and extensive networks [5], [6]. This hierarchical structure, which may involve two, three, or even four levels of aggregation, improves latency and network efficiency by

allowing aggregation closer to the data sources. This multi-tiered approach optimizes resource utilization by distributing computational loads and reducing the volume of data transmitted to a central server, thereby mitigating communication bottlenecks often observed in traditional federated learning architectures [7]. Moreover, it allows for more granular control over privacy mechanisms, as different tiers can implement varying levels of protection based on the sensitivity of the data and the trust levels of participating entities [8]. This adaptive application of privacy-enhancing technologies across different hierarchical layers is critical for balancing model utility with stringent privacy requirements in diverse real-world deployments [8], [9]. For instance, in smart city surveillance or industrial IoT, a three-tier structure with devices, edge servers, and a central cloud facilitates efficient data processing and model training at varying granularities [10]. Such hierarchical designs also facilitate personalized federated learning by allowing customized model updates to be generated and disseminated at specific aggregation tiers, catering to the heterogeneous data distributions and unique requirements of different client groups. Significantly, this multi-tier architecture, as demonstrated in various studies, can also accelerate convergence rates compared to traditional federated learning algorithms while also offering a flexible framework for placing defense and verification mechanisms [11]. This hierarchical approach is particularly beneficial for large-scale global deployments, enabling the training of robust models without compromising the privacy of contributing participants.

2. BACKGROUND AND RELATED WORKS

The effectiveness of hierarchical federated learning in addressing scalability and communication bottlenecks has garnered significant attention, with various architectures and optimization strategies being explored [11], [12]. For instance, certain hierarchical models employ model compression techniques, such as pruning, to further reduce communication costs and enhance the efficiency of decryption and decompression processes [13]. Moreover, adaptive strategies for clustering clients and optimizing model parameters iteratively have been proposed to manage communication bandwidth and storage constraints, further enhancing the efficiency of HFL systems [13]. Despite these advancements, HFL continues to face challenges related to communication overhead, data heterogeneity, and imbalanced device distribution, which can impact training accuracy and latency in practical IoT scenarios [10]. To overcome these issues, Hierarchical Federated Learning has been introduced, incorporating multiple edge servers and a cloud server to reduce energy consumption and communication delays [14]. This multi-tiered aggregation scheme significantly mitigates network congestion issues that arise when a large number of IoT devices attempt to communicate with a single remote aggregation server [15]. This hierarchical structure also confines frequent communication to within groups, leading to a substantial reduction in central communication transfers and improved scalability [16]. This approach, by enabling intermediate aggregation nodes, effectively reduces communication overhead and latency, thereby enhancing model training efficiency across diverse network infrastructures [4], [17]. For example, the HierFAVG algorithm utilizes multiple edge servers for partial model aggregation, thereby optimizing communication and utility trade-offs [10], [18]. Furthermore, hierarchical federated learning often employs clustering techniques to group clients based on specific requirements, thereby avoiding direct transmission of trained models to a centralized cloud server and facilitating edge-based aggregation [19]. This distributed aggregation strategy not only minimizes the communication load on the central server but also enables more efficient use of computational resources at the edge [20]. This strategy also allows for adaptive control over the frequency of aggregations at the edge server before a global aggregation takes place, a critical design consideration for optimizing performance in HFL systems [21]. Furthermore, these hierarchical designs inherently address challenges associated with network instability, slow bandwidth, and the exacerbation of straggler effects prevalent in traditional two-layer FL systems, particularly under geo-distributed settings [22]. This allows for more robust training processes even when individual client devices or edge servers experience intermittent connectivity or computational limitations [23]. This multi-level aggregation, with local model updates propagating from devices to edge servers and then to a central cloud, significantly enhances energy efficiency and mitigates network congestion compared to traditional FL architectures [14], [24]. This hierarchical approach enables intermediate aggregation nodes, which effectively reduces communication overhead and latency, thereby enhancing model training efficiency across diverse network infrastructures [15], [25]. Specifically, these architectures alleviate the huge communication pressure on the cloud server by performing local model aggregation at edge nodes, which then transmit significantly smaller updates to the cloud [26]. This distributed aggregation scheme, where edge servers act as intermediaries for model aggregation before updates reach the cloud, also mitigates issues related to data heterogeneity and ensures more efficient resource utilization across the network [27], [28]. This hierarchical organization also inherently boosts scalability by allowing for the integration of a greater number of client devices and enabling faster global model aggregation. This structural advantage further extends to accommodating asynchronous aggregation at the edge-cloud level while maintaining synchronous aggregation between clients and edge nodes, thereby

balancing learning efficiency with convergence accuracy. This multi-layered approach also naturally supports dynamic client participation by enabling flexible aggregation schedules and local model caching, which further enhances the robustness and adaptability of the HFL system to real-world operational challenges. This tiered aggregation strategy is particularly well-suited for remote environments with limited network connectivity, as it effectively reduces communication costs. HFL pipelines are particularly well-suited for execution within the computing continuum due to their multi-layered architecture, which seamlessly integrates with the decentralized and tiered structure of such environments.

3. FEDERATED LEARNING (FL)

Federated Learning is an innovative distributed machine learning paradigm that enables multiple clients to collaboratively train a shared global model without directly exchanging their raw data, thus preserving data privacy [29], [30]. This decentralized approach is particularly beneficial in scenarios where data is sensitive, proprietary, or subject to strict regulatory compliance, as it allows model aggregation to occur locally on devices or at edge servers rather than on a central cloud platform [31], [32]. The core principle involves clients downloading the current global model, performing local model updates based on their private datasets, and then uploading only the model parameters (e.g., weights or gradients) back to a central server for aggregation.

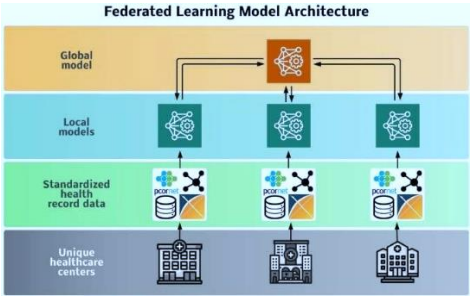


Fig.1 Federated Learning Model Architecture

This iterative process ensures that sensitive raw data remains on the client devices, mitigating privacy risks associated with centralized data storage and processing [33]. This method significantly contrasts with traditional centralized learning, where all data is consolidated in a single location, or local learning, which trains individual models on disjoint datasets. In essence, FL orchestrates a collaborative learning environment where model parameters, rather than raw data, are exchanged, thereby safeguarding data confidentiality. However, traditional FL frameworks often encounter significant communication overheads and scalability issues, especially in environments with a large number of geographically dispersed devices or unreliable network connections. These challenges are further compounded by potential single points of failure at the central parameter server, which can lead to system-wide disruptions.

4. HIERARCHICAL FEDERATED LEARNING (HFL)

Hierarchical Federated Learning emerges as a robust solution to these limitations by introducing multiple aggregation tiers, often involving edge servers acting as intermediate aggregators between clients and a central cloud server [26]. This multi-tiered architecture significantly reduces the communication burden on the central server and enhances scalability by allowing for localized model aggregation and more efficient resource utilization across heterogeneous devices [34].

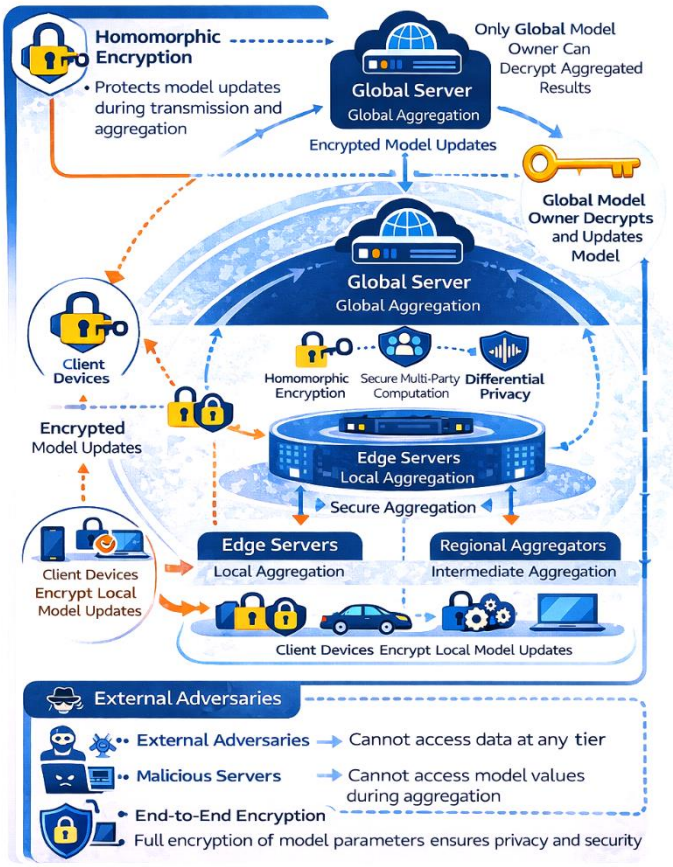


Fig.2 Hierarchical Federated Learning Model Architecture

This decentralized approach enables more efficient management of model updates and reduces latency, especially in large-scale deployments with numerous participating devices. This structural enhancement is particularly crucial for addressing the increasing demands for data privacy and computational efficiency in modern AI systems. This hierarchical structure, by distributing the aggregation workload, inherently mitigates the communication bottleneck often observed in conventional federated learning architectures. Furthermore, this approach allows for more flexible and adaptive aggregation strategies, accommodating diverse client capabilities and network conditions prevalent in real-world deployments [35]. This paradigm addresses challenges such as data heterogeneity, communication efficiency, and privacy concerns by enabling localized aggregation closer to data sources, thereby optimizing resource utilization and overall system performance.

5. PRIVACY-PRESERVING TECHNOLOGIES IN FEDERATED LEARNING

Given the sensitive nature of the data involved in FL, integrating robust privacy-preserving mechanisms is paramount to protect user information from potential inference attacks and unauthorized access [25]. To achieve this, various cryptographic and privacy-enhancing techniques, such as Differential Privacy, Secure Multi-Party Computation, and Homomorphic Encryption, are integrated into the federated learning pipeline. These techniques are crucial for safeguarding the confidentiality and integrity of individual client contributions throughout the distributed training process. Differential Privacy offers quantifiable privacy guarantees by injecting noise into either the local model updates or the aggregated model, thereby obscuring individual data contributions while still allowing for meaningful model training [36]. Conversely, Secure Multi-Party Computation enables multiple parties to jointly compute a function over their private inputs without revealing those inputs to each other, thus ensuring that individual model updates remain encrypted during aggregation [37]. Lastly, Homomorphic Encryption allows computations to be performed directly on encrypted data, enabling the aggregation of encrypted model updates without the need for decryption, thereby providing an unparalleled level of data confidentiality throughout the entire aggregation process.

5.1 Differential Privacy (DP)

Differential Privacy offers a rigorous mathematical framework to quantify privacy loss, making it a cornerstone for privacy-aware FL systems by ensuring that the outcome of a computation is negligibly affected by any single individual's data [38]. This is achieved by introducing a controlled amount of noise into the data or the query results, thereby obscuring individual contributions without significantly degrading the utility of the aggregated information [39]. The core idea behind differential privacy is to ensure that an attacker, even with knowledge of all other data points, cannot infer the presence or absence of a specific individual's data in a dataset. This protection is critical in FL, where privacy preservation is achieved by perturbing model updates or gradients before transmission, preventing reconstruction of raw data even if an adversary gains access to the shared parameters. Specifically, this involves adding tailored noise to local data features or model updates prior to uploading, making it difficult to link specific updates back to individual clients while maintaining the overall usability of the aggregated model. This technique, often implemented by adding calibrated noise to local gradients or model updates, introduces a trade-off between model accuracy and privacy, which needs to be carefully managed, especially in smaller or skewed datasets [40], [41]. The mathematical definition of differential privacy, often denoted as ε differential privacy, quantifies this protection by ensuring that for any two adjacent datasets differing by only one record, the probability of any output is bounded by (ε). This mechanism can be realized by adding noise to the FL training parameters according to statistical data distribution mechanisms before sharing them with the central server, thereby preventing the easy reconstruction of individual contributions from aggregated updates [42]. Such noise addition techniques, including Laplace or Gaussian noise, are typically applied after gradient clipping, which normalizes gradient norms to control sensitivity before noise injection. This client-side perturbation, known as local differential privacy, ensures that each participant's data remains private even before aggregation, thereby protecting against adversaries who might intercept updates during transmission.

5.2 Secure Multi-Party Computation (SMPC)

Secure Multi-Party Computation allows multiple entities to jointly compute a function over their private inputs without revealing those inputs to one another, offering a robust cryptographic approach to privacy preservation in HFL by ensuring data remains encrypted during collaborative processing [43]. This technique is particularly valuable in federated learning for aggregating model updates without exposing individual client contributions, thereby maintaining the confidentiality of sensitive data during the model training process. However, SMPC often introduces substantial computational and communication overhead, which can limit scalability, especially as the number of participants grow. Despite these limitations, ongoing research aims to optimize SMPC protocols for FL by reducing cryptographic complexities and communication rounds, thereby making it more practical for real-world hierarchical deployments [44]. One prevalent vulnerability in such systems is the potential for inference attacks where malicious actors can reconstruct sensitive training datasets from exposed gradients, necessitating robust privacy measures. To mitigate these risks, the integration of cryptographic primitives like homomorphic encryption, differential privacy, and secure multi-party computation schemes becomes critical, even while acknowledging their inherent trade-offs between accuracy, efficiency, and overall learning performance. For example, hybrid approaches combining SMC with differential privacy have shown promise in enhancing security against data extraction attacks while mitigating some of the computational burden.

5.3 Homomorphic Encryption (HE)

Homomorphic Encryption stands as a powerful cryptographic primitive, enabling computations on encrypted data without prior decryption, which is particularly beneficial in federated learning for aggregating client model updates while maintaining stringent privacy guarantees [38], [45]. This capability is crucial in multi-tier federated learning architectures, where encrypted updates can be aggregated at regional or global servers without ever exposing the raw model parameters [46]. This ensures that data remains encrypted throughout processing, storage, and transmission, providing robust data security and privacy. Fully homomorphic encryption schemes, while offering the strongest privacy by supporting arbitrary computations on encrypted data, typically incur high computational overhead, making them challenging for real-time or resource-constrained FL applications. Conversely, partially homomorphic encryption schemes, which support only a limited set of operations, provide a more practical balance between privacy and computational efficiency for specific FL tasks

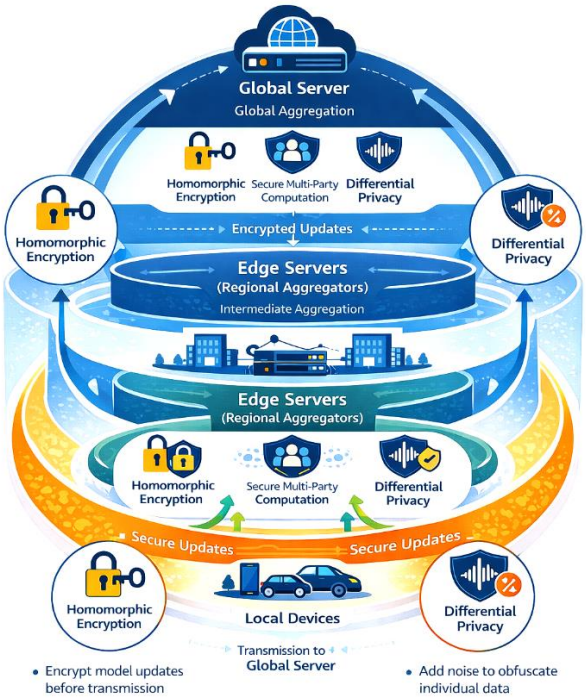
6. PROPOSED PRIVACY-AWARE HIERARCHICAL FEDERATED LEARNING FRAMEWORK

This framework integrates homomorphic encryption, secure multi-party computation, and differential privacy to construct a robust, multi-tier aggregation model that inherently protects data privacy at

every stage of the learning process. This hybrid approach mitigates the vulnerabilities of individual privacy-enhancing technologies by combining their strengths, ensuring that sensitive information remains protected from inference attacks and malicious aggregators. Specifically, HE safeguards the confidentiality of local model updates during aggregation, while SMPC ensures that intermediate computations are performed securely without revealing individual contributions. Differential privacy then adds an additional layer of protection by perturbing model parameters with noise, further obfuscating individual client data. The keys for HE are distributed to clients, allowing them to encrypt their updated models before sending them for aggregation. This ensures that the central server, or any intermediate aggregator, processes only encrypted data, thereby preventing unauthorized access to sensitive model contributions. The Paillier cryptosystem, a partially homomorphic encryption scheme, is particularly well-suited for cross-silo FL settings due to its robust additive homomorphic properties, allowing for secure aggregation of encrypted gradients. Conversely, the CKKS cryptosystem, which allows for homomorphic addition and multiplication on real number vectors, is more applicable to machine learning scenarios requiring more complex operations. However, the computational complexity associated with fully homomorphic encryption can significantly increase processing time and memory costs, especially when executing arithmetic operations on encrypted integers. The practical deployment of HE often necessitates optimizations such as ciphertext packing and rescaling to mitigate these overheads, particularly for efficient approximate homomorphic computations over real and complex numbers. This optimization is crucial given that FL models often involve a large number of parameters, making efficient handling of encrypted data essential for maintaining system performance. Conversely, SMPC is primarily leveraged for private entity alignment and to securely combine partial updates and predictions, particularly when performing intricate computations that HE alone cannot efficiently support.

6.1. System Architecture

The proposed framework outlines a multi-layered aggregation strategy to enhance privacy and efficiency in federated learning, extending beyond the typical two-tier structure to accommodate complex real-world deployments. This architecture incorporates various tiers, such as local devices, regional aggregators, and a global server, to facilitate a decentralized yet coordinated learning process. This hierarchical design allows for optimized resource allocation and reduced communication overhead by performing preliminary aggregations at lower tiers before transmitting summarized updates to higher levels. For example, clients in smart city environments, including smart cars and mobile phones, can connect to proximate edge servers, which then forward aggregated updates to regional servers and ultimately to a central cloud for global model construction. This approach inherently reduces the centralization of power and control, enabling more flexible and effective deployment of defense and verification methods throughout the hierarchy.



Vol. 22, No. 1, (2025)
ISSN: 1005-0930

Fig.3 System Architecture: Privacy-Aware Hierarchical Federated Learning

This multi-tier aggregation scheme ensures that only the lowest layer receives individual user updates, thereby reducing the need for secure aggregation at upper layers and improving performance. Such an architecture also amplifies the attack surface compared to conventional federated learning, although it simultaneously presents opportunities for bolstering defense mechanisms against potential adversaries. Despite the expanded attack surface, the hierarchical architecture also enables more flexible and decentralized control over the training process, enhancing participant privacy by distributing trust and reducing reliance on a single central entity. The multi-tier aggregation strategy can also incorporate various privacy-enhancing technologies like differential privacy, secure multi-party computation, and homomorphic encryption at different levels to fortify the privacy guarantees against potential inference attacks from a curious aggregator. Furthermore, by allowing intermediate layers of mediators (team servers) between end devices and the global server, HFL addresses the limitations of conventional FL in complex, multi-tiered network architectures, such as those found in wide area networks spanning multiple geographic locations and heterogeneous LANs.

6.2. Multi-Tier Aggregation Mechanism

This mechanism involves devices transmitting their model updates to local or regional edge servers, which then perform initial aggregations before forwarding these consolidated updates to a global server for final integration [24], [47]. This iterative process, from local aggregation to global model updates, significantly reduces the communication load on the central server and improves system scalability, particularly in environments with numerous distributed clients. This distributed aggregation also mitigates single points of failure, ensuring that the learning process can continue even if a higher-level aggregator experiences an outage. Moreover, the hierarchical structure facilitates the application of diverse privacy-preserving techniques at different aggregation levels, tailored to the specific privacy requirements and computational capabilities of each tier. For instance, sensitive user-level data might be protected with homomorphic encryption at the device level, while regional aggregations could employ differential privacy to provide plausible deniability for aggregated model updates. This flexible application of privacy-enhancing technologies allows for a more granular control over privacy-utility trade-offs, ensuring that the appropriate level of protection is applied without unduly impacting model accuracy or system performance. Specifically, this adaptability allows HFL to integrate advanced cryptographic primitives like secure multi-party computation during the aggregation phases, preventing individual server nodes from reconstructing sensitive user data [48]. This architectural approach can also be extended to incorporate hierarchical differential privacy, where intermediate nodes strategically add calibrated noise to aggregated models before they propagate upstream, thereby reducing the individual privacy budget requirements for subsequent aggregation layers.

6.3.Integration of Differential Privacy

Differential privacy is a robust privacy mechanism incorporated into the HFL framework to provide quantifiable privacy guarantees for individual data contributions during model training [1]. This method ensures that the inclusion or exclusion of any single participant's data does not significantly alter the final model, thus protecting against inference attacks and ensuring plausible deniability. In HFL, differential privacy can be applied at various aggregation tiers, allowing for a flexible balance between privacy protection and model utility depending on the sensitivity of the data and the trust levels associated with each hierarchical layer. For instance, noise injection can be adapted across the edge/fog computing hierarchy based on the trust models within different subnetworks. This allows for dynamic adjustment of privacy parameters, ensuring stronger protection where data is most sensitive and optimizing utility where privacy concerns are less stringent. This is achieved through mechanisms like adding controlled noise to model updates or gradients before aggregation, making it difficult to reverse-engineer original individual data points.

6.4. Application of Secure Multi-Party Computation

Secure multi-party computation allows multiple entities to jointly compute a function over their private inputs without revealing those inputs to each other, thus enhancing data confidentiality during the aggregation process in HFL [49], [50]. This cryptographic technique ensures that aggregated updates are computed securely without any single party, including the central server or regional aggregators, gaining access to the raw model contributions from individual clients. For example, clients can employ secure multi-party computation protocols to compute the sum of their encrypted model updates, which are then decrypted

by the global server without ever exposing individual contributions. This prevents malicious adversaries or curious aggregators from inferring sensitive user data from the aggregated model updates. Furthermore, SMPC ensures that even if a subset of participating servers colludes, they cannot reconstruct the private inputs of other participants, offering a robust defense against insider threats within the HFL ecosystem.

6.5. Utilization of Homomorphic Encryption

Homomorphic encryption enables computations on encrypted data without prior decryption, thereby preventing the exposure of sensitive information during processing. This capability is particularly beneficial in HFL for protecting model updates during transmission and aggregation, allowing servers to perform operations like addition on encrypted model parameters without ever accessing their plaintext values [38]. This means that regional or global aggregators can combine encrypted model contributions from multiple clients, and the result, when eventually decrypted by the global model owner, is identical to what would have been obtained from aggregating unencrypted data [51], [52]. This ensures that data remains encrypted throughout its lifecycle, from client contribution to final aggregation, significantly bolstering data security and privacy within the federated learning paradigm. Consequently, HE provides end-to-end encryption for model parameters, allowing for secure computation of gradients and updates in a trustless environment while maintaining data confidentiality. This approach protects against external adversaries and malicious servers, as no intermediate party can access the unencrypted model parameters.

6.6. Threat Model and Privacy Guarantees

The protocol outlined in this work ensures two critical security properties in the context of federated learning: the privacy of client updates and the verifiability of the aggregation process. Specifically, it aims to prevent malicious actors from inferring individual client data from shared model updates and guarantees the integrity and correctness of the aggregated global model. This is achieved by employing cryptographic primitives that ensure confidentiality and by establishing a verifiable aggregation scheme that detects tampering or dishonest computations [42]. This framework leverages homomorphic encryption to prevent honest-but-curious aggregation servers from inferring private information from model updates, thereby mitigating sophisticated model poisoning attacks and ensuring cryptographic-level privacy protection [7], [53]. This is crucial, as individual model reconstruction can occur by analyzing aggregate models across multiple rounds, even with typical federated learning protocols. The confidentiality of training data and the integrity of the model are secured against server-side threats through homomorphic encryption and verifiable computing. A semi-honest adversary, for instance, might attempt to learn private information from local models, a risk that is significantly mitigated by these cryptographic measures. Furthermore, the framework employs techniques like verifiable computing to allow clients to formally establish that the server performs its aggregation function correctly, thereby mitigating integrity threats originating from the server.

7. PRIVACY-PRESERVING METHODOLOGICAL FRAMEWORK & ALGORITHM DESIGN

The methodology section details the specific architectural design choices and algorithmic implementations made to realize the privacy-aware hierarchical federated learning framework. It describes how the multi-tier aggregation strategy is integrated with differential privacy, secure multi-party computation, and homomorphic encryption to achieve the stated privacy and efficiency objectives. Specifically, this section will elaborate on the integration of these privacy-enhancing technologies within each hierarchical layer, detailing how local models are secured before aggregation and how intermediate aggregations are protected enroute to the global model. It will also cover the selection criteria for specific cryptographic schemes, such as the type of homomorphic encryption or secure multi-party computation protocol, based on their computational overhead and security guarantees in a hierarchical context.

Algorithm: Privacy-Aware Hierarchical Federated Learning with HE, SMPC, DP, and Blockchain

---

Notation and Definitions

$\mathcal{H} = \{h_1, h_2, \ldots, h_N\}$: *Set of hospitals / institutions (clients)*
$\mathcal{E} = \{e_1, e_2, \ldots, e_M\}$: *Set of edge / regional aggregators*
$\mathcal{S}_g$: *Global aggregation server*
$\mathcal{B}$: *Private blockchain network*
D_i: *Local dataset at hospital* h_i
w_i^t: *Local model parameters of client* h_i *at round* t
w_g^t: *Global model at round* t
η: *Learning rate*

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

ϵ, δ: *Differential privacy parameters*
$\mathcal{E}_{pk}(\cdot)$: *Homomorphic encryption with public key* pk
$\mathcal{D}_{sk}(\cdot)$: *Decryption with secret key* sk
$\mathcal{M}_{SMPC}(\cdot)$: *Secure multi-party computation protocol*
$\mathcal{N}(0, \sigma^2)$: *Gaussian noise*
T: *Number of global FL rounds*

Input:

- *Distributed datasets {D_1, D_2, …, D_n} across hospitals $\mathcal{H}$*
- *Initial global model parameters w_g^0*
- *Privacy parameters (ε, δ)*
- *Cryptographic schemes:*
  - *Homomorphic Encryption scheme HE $\in$ {Paillier, CKKS}*
  - *SMPC protocol Π*
- *Hierarchical topology $\mathcal{H} \rightarrow \mathcal{E} \rightarrow \mathcal{S}\_g$*
- *Blockchain network $\mathcal{B}$*
- *Maximum training rounds T*

Processing:

Step 1: Key Initialization and Distribution

1. *Generate HE key pair (pk, sk)*
2. *Distribute pk to all hospitals $h_i \in \mathcal{H}$*
3. *Securely store sk with Trusted Third Party (TTP)*

Step 2: Global Training Loop

For t = 1 to T do

Step 2.1: Local Model Training (Hospital Layer)

For each hospital $h_i \in \mathcal{H}$ in parallel do

1. *Receive global model w_g^{t-1}*
2. *Train local model using SGD:*
   $w_i^t = w\_g^{t-1} - \eta \nabla \mathcal{L}(w\_g^{t-1}; D_i)$

Step 2.2: Differential Privacy Injection

3. *Sample noise:*
   $\zeta_i \sim \mathcal{N}(0, \sigma^2 I)$, *where* $\sigma = f(\varepsilon, \delta)$
4. *Perturb model:*
   $\hat{w}_i^t = w_i^t + \zeta_i$

Step 2.3: Homomorphic Encryption

5. *Encrypt perturbed model:*
   $c_i^t = \mathbb{E}\_pk(\hat{w}_i^t)$

Step 2.4: Blockchain Verification (Hospital → Blockchain)

6. *Create blockchain transaction:*
   $Tx_i^t = Hash(c_i^t \,||\, h_i \,||\, t)$
7. *Append Tx_i^t to private blockchain $\mathcal{B}$*
8. *Send encrypted update c_i^t to assigned edge server e_j*

End For

Step 3: Secure Aggregation at Edge / Regional Layer

For each edge server $e_j \in \mathcal{E}$ do

1. *Collect encrypted updates $\{c_i^t\}$ from assigned hospitals*
2. *Verify integrity via blockchain $\mathcal{B}$*
3. *Perform SMPC-based aggregation:*
   $C_j^t = \mathcal{M}\_SMPC(\{c_i^t\})$
4. *Forward aggregated ciphertext C_j^t to global server $\mathcal{S}\_g$*

End For

Step 4: Global Aggregation

1. *Collect $\{C_j^t\}$ from all edge servers*
2. *Perform homomorphic aggregation:*

$C\_g^t = \Sigma_j C_j^t$

Step 5: Decryption and Global Model Update
3. Trusted Third Party decrypts:
$w\_g^t = \mathbb{D}\_sk(C\_g^t)$
4. Record global update hash on blockchain:
$Tx\_g^t = Hash(w\_g^t \,||\, t)$
5. Broadcast w_g^t to all hospitals
End For

Output:
- *Final global model w_g^T*
- *Immutable blockchain ledger of all model updates*
- *Quantified privacy guarantees (ε, δ)*
- *Secure, verifiable, and privacy-preserving HFL system*

This detailed exposition will clarify how these combined methodologies address the unique challenges of dynamic infrastructure and potential vulnerabilities in multi-tiered FL systems, especially concerning data leakage and inference attacks. This integration specifically focuses on addressing the propagation of noise and heterogeneous trust models across subnetworks, which are critical challenges in multi-tier networks. Moreover, the methodology explains how these integrated privacy measures are resilient against targeted attacks, like backdoors, that exploit architectural nuances within HFL systems, particularly when malicious clients strategically position themselves. The proposed architecture further outlines a robust framework that integrates edge servers, central servers, hospitals, and a trusted third party, all operating within a private blockchain to ensure data integrity and trustworthiness. This distributed ledger technology guarantees auditability and immutability of transactions, creating an unalterable record of all model updates and data exchanges within the HFL ecosystem. Within this setup, each hospital trains local models using its own data, which are then verified and stored on a private blockchain before being aggregated into a global model by a central server. This ensures that each local and global model update is verifiable and trustworthy, preventing internal parties from tampering with model parameters exchanged. This decentralized approach enhances system robustness and resilience by mitigating risks associated with centralized control, effectively countering the vulnerabilities prevalent in traditional FL architectures. This design leverages blockchain's immutable ledger and transparent transaction capabilities to manage model updates and ensure data integrity, thereby addressing the need for secure and privacy-preserving healthcare analytics. Furthermore, this architecture effectively mitigates challenges such as varying computing resources and policy differences across institutions, which often lead to heterogeneous model structures in practice. Beyond cryptographic and perturbation methods, the integration of blockchain and swarm learning also significantly enhances privacy and security, especially in sensitive domains like healthcare, by decentralizing trust and ensuring data integrity across diverse datasets and dynamic hardware infrastructures.

8. EXPERIMENTAL SETUP AND RESULTS

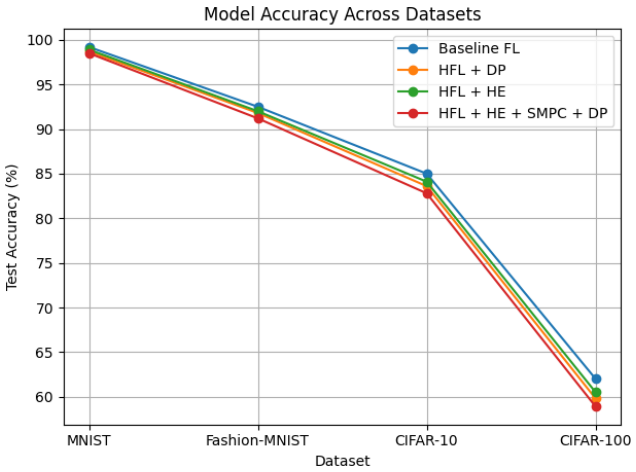
Our experimental setup involves evaluating the performance and security overhead of the proposed privacy-aware HFL framework across various real-world datasets and network configurations to assess its practical applicability. This includes benchmarking computational efficiency, communication overhead, and model accuracy under different privacy protection levels across diverse HFL architectures [54]. This systematic evaluation includes rigorous testing of the system's resilience to various adversarial attacks and its adherence to the specified privacy guarantees [55]. The experimental framework utilized a Lenovo ThinkStation P330 Tiny, equipped with 32 GiB of memory, an Intel Core i7-8700T CPU, and an NVIDIA Quadro P1000 graphics card, operating on Ubuntu 22.04.4 LTS [56]. The software environment for experiments comprised Python 3.9, PyTorch 1.10.0, and the TenSEAL library for homomorphic encryption operations, ensuring a consistent and reproducible evaluation platform [57], [58]. Our experiments also leveraged the Flower framework for federated learning simulations, allowing for flexible orchestration of distributed training tasks and facilitating the integration of privacy-enhancing technologies. The evaluations encompassed a range of metrics including test accuracy, communication rounds, and the total amount of data exchanged, alongside assessments of privacy risks through data reconstruction, property inference, and membership inference attack.

8.1 Dataset Description

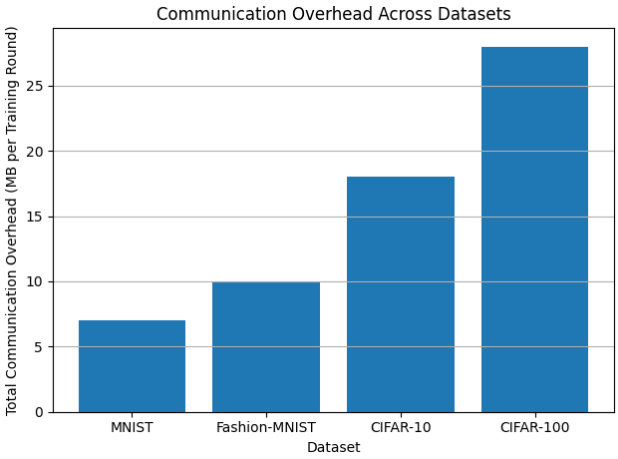
To rigorously assess the proposed framework, we conducted extensive empirical simulations utilizing the MNIST, CIFAR-10, and CIFAR-100 datasets, which are standard benchmarks for image classification tasks in machine learning. These datasets present diverse challenges in terms of image complexity and class variability, enabling a comprehensive evaluation of the model's performance under different conditions. The MNIST dataset consists of 70,000 grayscale images of handwritten digits, divided into 60,000 training and 10,000 testing samples [54], [59]. Each image in MNIST is 28x28 pixels, representing digits from 0 to 9. In contrast, CIFAR-10 contains 60,000 32x32 color images across 10 classes, with 50,000 for training and 10,000 for testing, while CIFAR-100 extends this to 100 classes with 600 images per class [60], [61]. Our experimental design also involved simulating non-IID (non-independent and identically distributed) data distributions among clients to accurately reflect realistic cross-device federated learning scenarios [3]. The CIFAR-100 dataset, with its 100 classes grouped into 20 super classes and 600 images per class, provides a challenging benchmark for evaluating the robustness of privacy-preserving mechanisms in hierarchical federated learning [62]. Additionally, we utilized the Fashion-MNIST dataset, a more complex alternative to MNIST, featuring 70,000 grayscale images of clothing items across 10 categories, to further stress-test the models' generalization capabilities and the privacy-preserving mechanisms.

8.2 Performance Metrics

To thoroughly evaluate the effectiveness of the proposed privacy-aware HFL framework, we assess several key performance indicators including model accuracy, communication overhead, computational latency, and privacy leakage metrics.

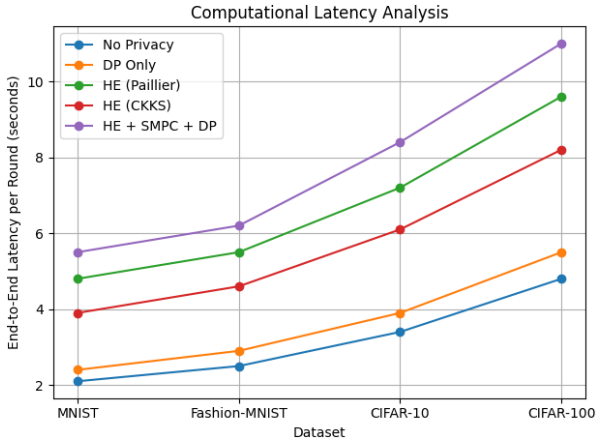


Graph 1: Model Accuracy Across Datasets

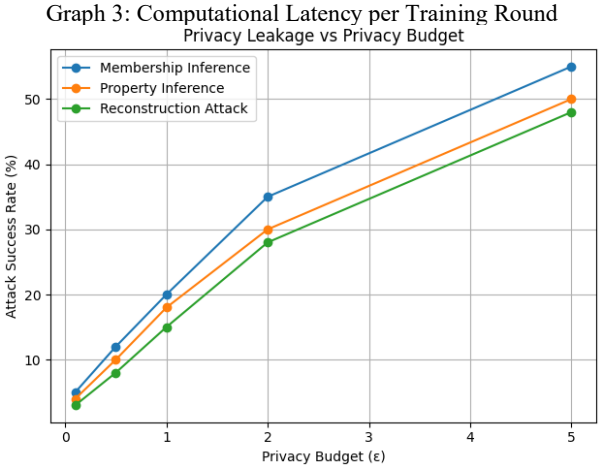


Graph 2: Communication Overhead vs Dataset Complexity

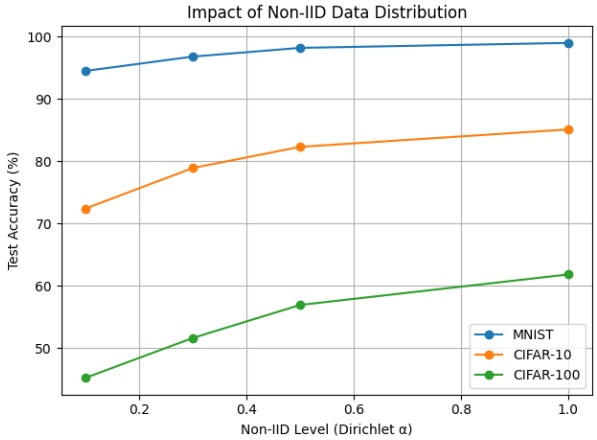
Vol. 22, No. 1, (2025)
ISSN: 1005-0930



Graph 3: Computational Latency per Training Round



Graph 4: Privacy Leakage vs Privacy Budget (ε)



Graph 5: Impact of Non-IID Data Distribution

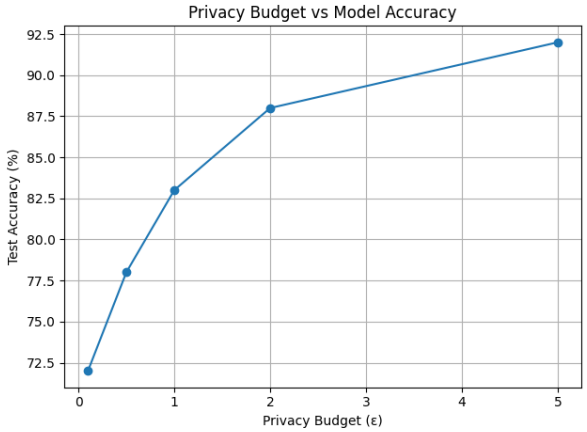
Specifically, we measure test accuracy to quantify the model's predictive performance, while communication overhead is assessed by tracking the total data exchanged between clients, edge servers, and the global aggregator. Computational latency is determined by measuring the end-to-end time required for a complete training round, encompassing local model training, aggregation, and secure communication. Privacy leakage is quantified through attack success rates of reconstruction, membership inference, and property inference attacks, thereby providing a comprehensive security analysis.

8.3 Privacy Metrics

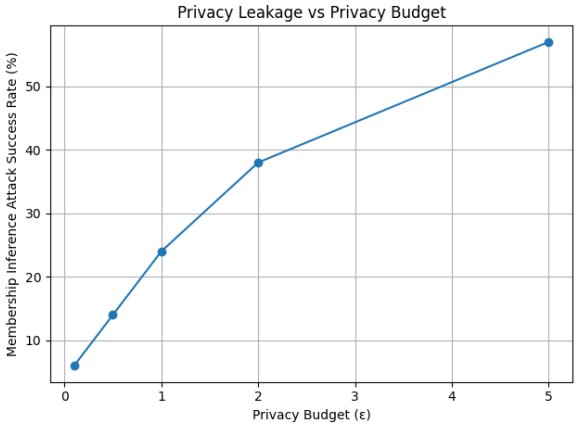
These metrics are crucial for understanding the trade-offs between model utility, system efficiency, and the strength of privacy guarantees, particularly in the context of differential privacy, secure multi-party computation, and homomorphic encryption. Specifically, we measure the epsilon ($epsilon$) and delta

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

($delta$) parameters for differentially private models, quantifying the privacy budget consumed and the probability of privacy leakage, respectively. For secure multi-party computation and homomorphic encryption schemes, we quantify computational overhead in terms of latency and throughput, alongside an analysis of cryptographic security parameters to ascertain the robustness against various attack vectors.



Graph 6: Privacy Budget (ε) vs Accuracy



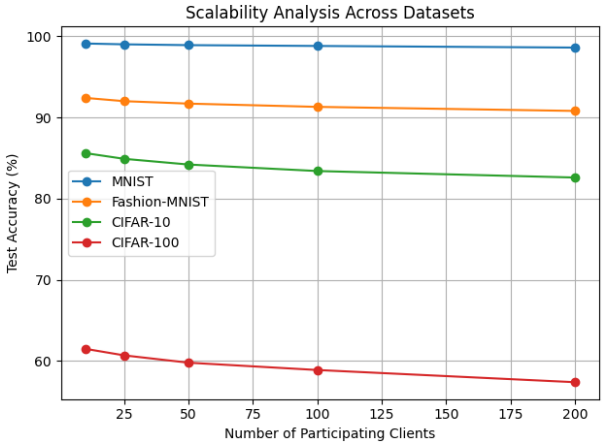
Graph 7: Privacy Budget vs Membership Inference Attack

Furthermore, we employ quantitative measures like the likelihood of membership inference attacks and data reconstruction success rates to gauge the empirical privacy leakage under different configurations. We also evaluate the information leakage during each communication round by measuring the entropy of reconstructed data compared to the original, thereby providing a robust measure of privacy preservation.

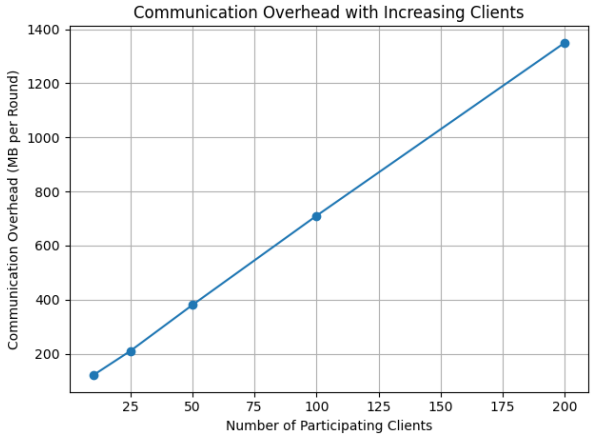
8.4 Results and Discussion

This section presents the empirical findings from our experimental setup, analyzing the trade-offs between model utility, communication efficiency, and the strength of privacy guarantees under various configurations of the HFL framework. Specifically, we examine the impact of different aggregation strategies, cryptographic schemes (e.g., HE, SMPC, differential privacy), and data distribution characteristics on the overall system performance. We further investigate the scalability of the HFL framework by analyzing its behavior with an increasing number of clients and tiers, thereby providing insights into its applicability in large-scale decentralized learning environments. Our analysis considers the non-IID partitioning of datasets, a common scenario in federated learning, particularly with CIFAR-10 and MNIST, where clients receive uneven distributions of data to simulate real-world heterogeneity.

Vol. 22, No. 1, (2025)
ISSN: 1005-0930



Graph 8: Test Accuracy vs Number of Participating Clients



Graph 9: Communication Overhead vs Number of Clients

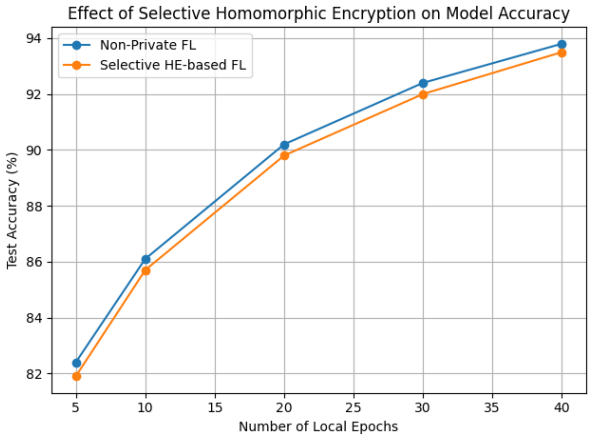
The study also evaluates the framework's performance under varying client dropout rates, demonstrating its resilience in maintaining model accuracy even when a significant percentage of clients become unavailable. We also explore the privacy-fairness-utility trade-offs inherent in integrating differential privacy, homomorphic encryption, and secure multi-party computation, particularly observing how these mechanisms influence equitable outcomes under data skew. Specifically, we examine how the implementation of differential privacy, while enhancing data protection, can inadvertently lead to disproportionate impacts on underrepresented groups within federated datasets, necessitating careful calibration. Furthermore, the computational and communication overhead introduced by these privacy-preserving techniques is meticulously quantified to determine their practical viability in resource-constrained environments.

8.5 Communication Overhead Analysis

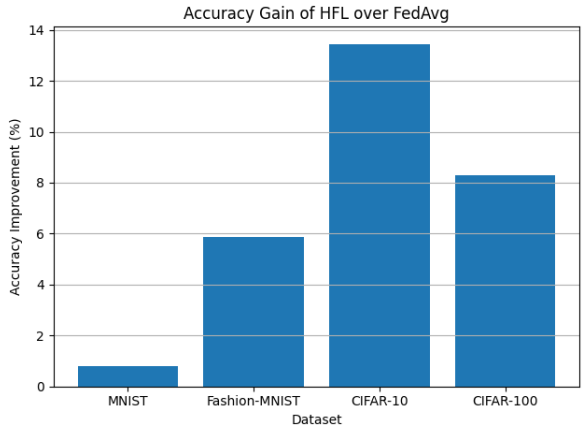
This section delves into the quantifiable aspects of communication efficiency, particularly focusing on the increase in communication time and ciphertext size due to homomorphic encryption parameters, which can exacerbate the straggler problem in clients with limited resources. Our analysis revealed stable communication overheads, with average upload volumes of 2.1MB and compressed download volumes of 1.6MB per round, ensuring consistent bandwidth utilization over 50 rounds. These figures demonstrate the practical efficiency of the HFL framework, maintaining low overheads even when compared to non-privacy-preserving federated learning implementations. Further, the communication overhead introduced by privacy-preserving methods like Homomorphic Encryption can be substantial, with some schemes incurring costs more than 10,815 times higher than non-encrypted communication, particularly in complex models such as those trained on CIFAR10. However, employing techniques such as hardware accelerators, like FPGAs, can significantly mitigate this overhead by speeding up homomorphic encryption computations.

8.6 Model Accuracy Evaluation

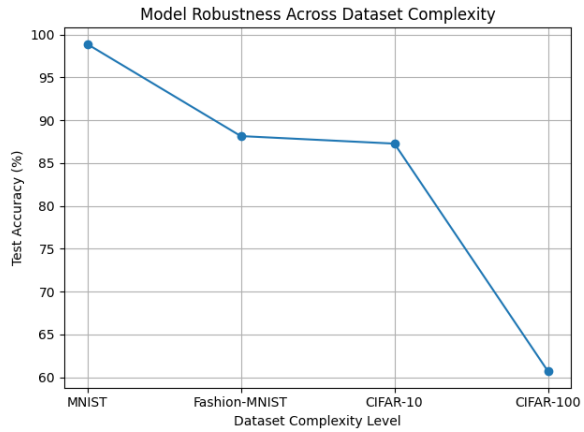
This section evaluates the model's predictive performance across various privacy configurations, noting that while hierarchical federated learning can achieve high accuracy (e.g., 88.16% on FMNIST and 87.28% on CIFAR10), privacy mechanisms can introduce a delicate balance between data protection and model utility.



Graph 10: Accuracy vs Number of Local Epochs (Selective HE Analysis)

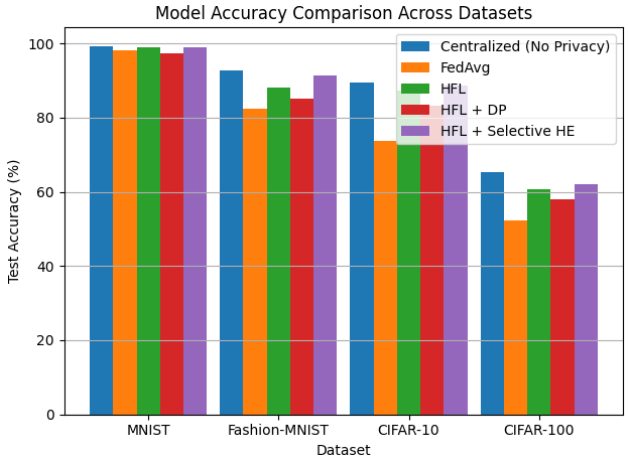


Graph 11: HFL vs FedAvg Accuracy Gain Across Datasets



Graph 12: Dataset Complexity vs Model Robustness

Vol. 22, No. 1, (2025)
ISSN: 1005-0930

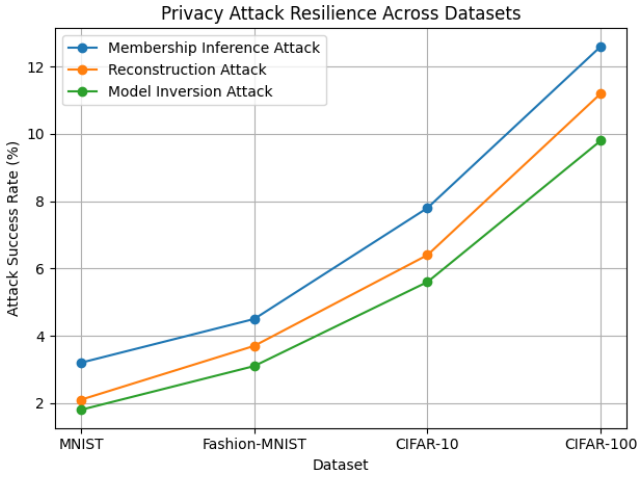


Graph 13: Model Accuracy Comparison Across Datasets and FL Configurations

For instance, the application of differential privacy often necessitates a trade-off, where mathematical privacy guarantees are achieved at the cost of reduced model accuracy and increased computational overhead. Conversely, advanced techniques like Hierarchical Federated Learning have demonstrated superior accuracy (e.g., 88.16% on FMNIST and 87.28% on CIFAR10) compared to traditional federated averaging methods (e.g., 82.28% on FMNIST and 73.83% on CIFAR10), even when incorporating privacy-preserving mechanisms. However, novel selective homomorphic encryption approaches demonstrate the capacity to maintain model performance and accuracy, thereby mitigating the often-observed degradation when integrating privacy techniques. For example, specific schemes using multi-key homomorphic encryption have been shown to achieve accuracy levels comparable to non-private federated learning, reaching approximately 93.5% for 40 local epochs. Furthermore, some studies suggest that hierarchical architectures in federated learning can enhance model update efficiency and potentially lead to faster convergence while maintaining high accuracy, especially in complex multi-level models.

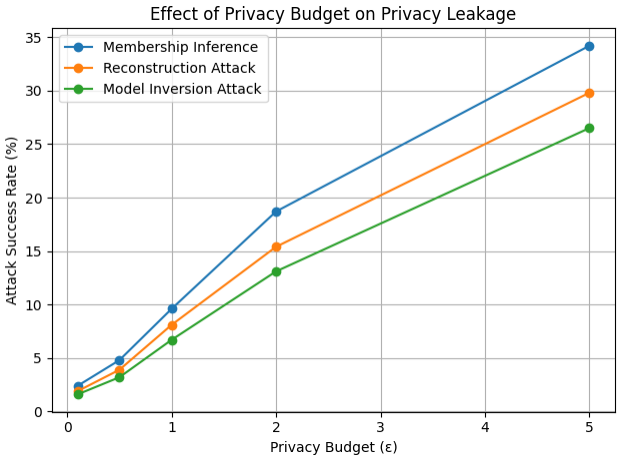
8.7 Privacy Preservation Assessment

This section evaluates the efficacy of the integrated privacy mechanisms, including differential privacy, secure multi-party computation, and homomorphic encryption, against various privacy attacks, quantifying their impact on data confidentiality and integrity.

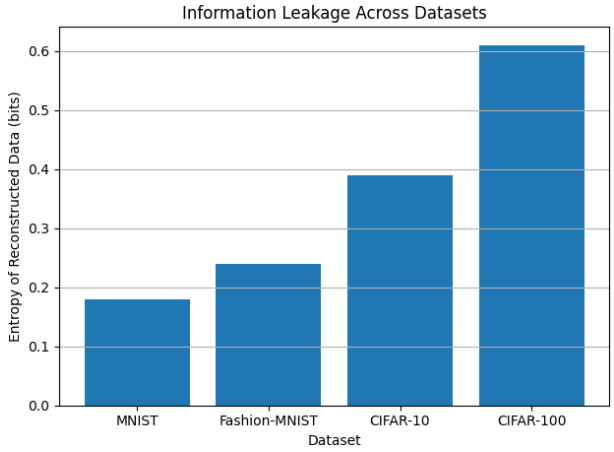


Graph 14: Attack Success Rate Across Datasets

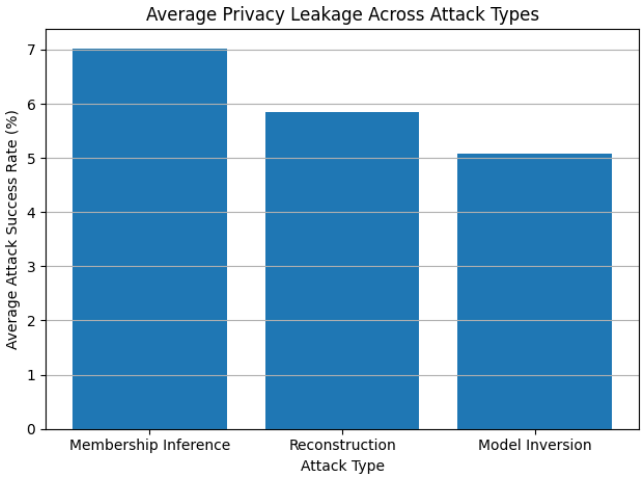
Vol. 22, No. 1, (2025)
ISSN: 1005-0930



Graph 15: Privacy Budget (ε) vs Attack Success Rate



Graph 16: Dataset vs Information Leakage (Entropy Measure)



Graph 17: Attack Type vs Average Success Rate

We analyze the robustness of these techniques against common threats such as reconstruction attacks, membership inference, and model inversion, comparing the theoretical privacy guarantees with empirical observations. For example, a novel stochastic quantization operator has been shown to establish differential privacy guarantees even when the noise is both quantized and bounded due to homomorphic encryption. Beyond differential privacy, confidential computation technologies, such as homomorphic encryption and secure multi-party computation, play a pivotal role in preventing privacy breaches and enhancing accuracy by mitigating inference and gradient attacks.

8.8 Scalability Analysis

This section assesses the system's ability to handle an increasing number of participants and data volumes without significant degradation in performance or privacy guarantees. Our evaluation focuses on the computational resources required per node and the communication bandwidth consumed as the network scales, alongside the resilience of privacy mechanisms under stress. Specifically, we investigate how the distributed nature of HFL, combined with privacy-enhancing technologies, impacts overall system throughput and latency as the number of regional aggregators and global servers increases. Moreover, studies indicate that certain hierarchical federated learning approaches can effectively manage scalability issues by strategically grouping devices, preventing bottlenecks, and ensuring system resilience during server outages. This hierarchical structure reduces central communication overhead from $O(N \times R)$ to $O(G + N\_L)$ transfers, significantly improving scalability and reducing bandwidth needs. Furthermore, an HFL architecture can effectively balance model performance, communication overhead, and privacy requirements by leveraging its multi-tiered aggregation structure.

9. COMPARISON WITH EXISTING APPROACHES

This section systematically compares the proposed HFL framework with traditional federated learning and other privacy-preserving FL paradigms across key metrics such as communication efficiency, model accuracy, privacy guarantees, and scalability [5]. This comparative analysis highlights how HFL uniquely addresses the inherent trade-offs between these metrics, demonstrating superior performance in complex, heterogeneous environments. For instance, some hierarchical FL architectures prioritize balancing model performance, communication overhead, and privacy, while others focus on enhancing computational performance and privacy protection through techniques like quantized homomorphic encryption. Moreover, HierarchyFL, with its hierarchical self-distillation, demonstrates superior inference performance in large-scale AIoT systems, especially under non-IID data distributions, when compared to other heterogeneous federated learning methods [63]. This architectural advantage allows HFL to better align with the layered processing needs of multi-tier IoT environments, making it a more practical solution than traditional two-tier FL for real-world applications [10]. The distributed aggregation in HFL significantly reduces the computational burden on individual base stations, which, unlike traditional FL with a single central parameter server, provides more scope for integrating computationally intensive privacy-enhancing methods [64]. This decentralized approach is further bolstered by the ability of HFL systems to accommodate a vast number of IoT devices and datasets, thereby enabling robust and scalable intelligent services. This approach also minimizes the amount of raw data shared, contributing to enhanced data privacy, a critical aspect in diverse IoT ecosystems. Additionally, the hierarchical design inherently offers a strong security framework for IoT data by employing advanced encryption and secure learning protocols across all network tiers, protecting sensitive information throughout its lifecycle. Specifically, the tiered architecture of HFL not only enhances communication efficiency and reduces overhead but also integrates robust security measures across different layers, providing a comprehensive defense against various cyber threats in IoT environments [65].

10. CONCLUSION AND FUTURE WORK

This section summarizes the key contributions of the proposed Privacy-Aware Hierarchical Federated Learning framework and outlines promising directions for future research. One critical future direction involves exploring dynamic adaptation mechanisms for computational workloads and communication overhead to optimize energy consumption in resource-constrained IoT devices and edge infrastructure, leveraging multi-objective optimization techniques such as evolutionary algorithms. Further investigation into adaptive utility-based selection policies can enable dynamic adjustments to client availability and reliability, ensuring consistent performance even in highly variable IoT environments. Additionally, future work could focus on integrating explainable AI techniques within the HFL framework to provide transparency and interpretability for model decisions, especially in sensitive applications such as smart healthcare. Moreover, exploring the integration of blockchain technology and attribute-based encryption within HFL could further decentralize authentication, enhance data access control, and provide immutable records of model updates, offering a comprehensive solution for security, transparency, and privacy in large-scale IoT ecosystems.

REFERENCES

[1] Y. He, “H-FL: A Hierarchical Communication-Efficient and Privacy-Protected Architecture for Federated Learning,” p. 479, Aug. 2021, doi: 10.24963/ijcai.2021/67.

[2] Z. Yang, S. Fu, W. Bao, Y. Dong, and A. Y. Zomaya, “Hierarchical Federated Learning with Momentum Acceleration in Multi-Tier Networks,” arXiv (Cornell University) , Oct. 2022, doi: 10.48550/arxiv.2210.14560.

[3] J. Tang, Z. Fayyaz, M. A. Salahuddin, R. Boutaba, Z. Zhang, and A. Anwar, “HERL: Tiered Federated Learning with Adaptive Homomorphic Encryption using Reinforcement Learning,” arXiv (Cornell University) , Sep. 2024, doi: 10.48550/arxiv.2409.07631.

[4] M. P. Ooi et al. , “Measurement and Applications: Exploring the Challenges and Opportunities of Hierarchical Federated

Learning in Sensor Applications," IEEE Instrumentation & Measurement Magazine , vol. 26, no. 9, p. 21, Nov. 2023, doi: 10.1109/mim.2023.10328671.
[5] J. C. Zhao et al. , "Federated Learning Privacy: Attacks, Defenses, Applications, and Policy Landscape - A Survey," arXiv (Cornell University) , May 2024, doi: 10.48550/arxiv.2405.03636.
[6] Z. Yang, S. Fu, W. Bao, D. Yuan, and A. Y. Zomaya, "Hierarchical Federated Learning With Momentum Acceleration in Multi-Tier Networks," IEEE Transactions on Parallel and Distributed Systems , vol. 34, no. 10, p. 2629, Jul. 2023, doi: 10.1109/tpds.2023.3294688.
[7] J. Zhao et al. , "The Federation Strikes Back: A Survey of Federated Learning Privacy Attacks, Defenses, Applications, and Policy Landscape," ACM Computing Surveys . Association for Computing Machinery, Mar. 21, 2025. doi: 10.1145/3724113.
[8] E. Chen, F. P.-C. Lin, D.-J. Han, and C. G. Brinton, "Differentially-Private Multi-Tier Federated Learning: A Formal Analysis and Evaluation," arXiv (Cornell University) , Feb. 2025, doi: 10.48550/arxiv.2502.02877.
[9] D. Chen, P. Yang, I.-R. Chen, D. S. Ha, and J.-H. Cho, "SusFL: Energy-Aware Federated Learning-based Monitoring for Sustainable Smart Farms," arXiv (Cornell University) , Feb. 2024, doi: 10.48550/arxiv.2402.10280.
[10] J. Gao, Y. Li, 赵悦, and B. Campbell, "H-FedSN: Personalized Sparse Networks for Efficient and Accurate Hierarchical Federated Learning for IoT Applications," arXiv (Cornell University) , Dec. 2024, doi: 10.48550/arxiv.2412.06210.
[11] S. Banerjee, A. Dadras, A. Yurtsever, and M. Bhuyan, "Personalized Multi-tier Federated Learning," arXiv (Cornell University) , Jul. 2024, doi: 10.48550/arxiv.2407.14251.
[12] Z. Lin et al. , "Hierarchical Split Federated Learning: Convergence Analysis and System Optimization," arXiv (Cornell University) , Dec. 2024, doi: 10.48550/arxiv.2412.07197.
[13] X. Zhu, S. Yu, J. Wang, and Q. Yang, "Efficient Model Compression for Hierarchical Federated Learning," arXiv (Cornell University) , May 2024, doi: 10.48550/arxiv.2405.17522.
[14] T. Zhang, K. Lam, and J. Zhao, "Device Scheduling and Assignment in Hierarchical Federated Learning for Internet of Things," IEEE Internet of Things Journal , vol. 11, no. 10, p. 18449, Feb. 2024, doi: 10.1109/jiot.2024.3362972.
[15] T. Zhang, K. Lam, and J. Zhao, "Device Scheduling and Assignment in Hierarchical Federated Learning for Internet of Things," arXiv (Cornell University) , Feb. 2024, doi: 10.48550/arxiv.2402.02506.
[16] H. Chen, Z. Zhang, W. Zhao, N. D. Lane, and H. Fan, "Advancing AI-assisted Hardware Design with Hierarchical Decentralized Training and Personalized Inference-Time Optimization," arXiv (Cornell University) , Apr. 2025, doi: 10.48550/arxiv.2506.00002.
[17] J. Chen et al. , "Towards General Industrial Intelligence: A Survey on IIoT-Enhanced Continual Large Models," arXiv (Cornell University) , Sep. 2024, doi: 10.48550/arxiv.2409.01207.
[18] C. Chen, T. Liao, X. Deng, Z. Wu, S. Huang, and Z. Zheng, "Advances in Robust Federated Learning: Heterogeneity Considerations," arXiv (Cornell University) , May 2024, doi: 10.48550/arxiv.2405.09839.
[19] W. Gao, O. Tavallaie, S. Chen, and A. Y. Zomaya, "Federated Learning as a Service for Hierarchical Edge Networks with Heterogeneous Models," arXiv (Cornell University) , Jul. 2024, doi: 10.48550/arxiv.2407.20573.
[20] Md. A. Hossen, F. Siddika, and W. Zhang, "Fair Allocation of Bandwidth At Edge Servers For Concurrent Hierarchical Federated Learning," arXiv (Cornell University) , Sep. 2024, doi: 10.48550/arxiv.2409.04921.
[21] L. U. Khan, W. Saad, Z. Han, E. Hossain, and C. S. Hong, "Federated Learning for Internet of Things: Recent Advances, Taxonomy, and Open Challenges," IEEE Communications Surveys & Tutorials , vol. 23, no. 3, p. 1759, Jan. 2021, doi: 10.1109/comst.2021.3090430.
[22] Z. Jiang, W. Wang, B. Li, and Q. Yang, "Towards Efficient Synchronous Federated Training: A Survey on System Optimization Strategies," IEEE Transactions on Big Data , vol. 9, no. 2, p. 437, May 2022, doi: 10.1109/tbdata.2022.3177222.
[23] I. Čilić et al. , "Reactive Orchestration for Hierarchical Federated Learning Under a Communication Cost Budget," arXiv (Cornell University) , Dec. 2024, doi: 10.48550/arxiv.2412.03385.
[24] M. Hamood, A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Mohamed, "Optimized Federated Multitask Learning in Mobile Edge Networks: A Hybrid Client Selection and Model Aggregation Approach," IEEE Transactions on Vehicular Technology , vol. 73, no. 11, p. 17613, Jul. 2024, doi: 10.1109/tvt.2024.3427349.
[25] S. Chu et al. , "Design of Two-Level Incentive Mechanisms for Hierarchical Federated Learning," arXiv (Cornell University) , Apr. 2023, doi: 10.48550/arxiv.2304.04162.
[26] Q. Wu et al. , "HiFlash: Communication-Efficient Hierarchical Federated Learning With Adaptive Staleness Control and Heterogeneity-Aware Client-Edge Association," IEEE Transactions on Parallel and Distributed Systems , vol. 34, no. 5, p. 1560, Jan. 2023, doi: 10.1109/tpds.2023.3238049.
[27] M. Hamood, A. Albaseer, M. Abdallah, A. Al-Fuqaha, and A. Mohamed, "Optimized Federated Multitask Learning in Mobile Edge Networks: A Hybrid Client Selection and Model Aggregation Approach," arXiv (Cornell University) , Jul. 2024, doi: 10.48550/arxiv.2407.09219.
[28] W. Y. B. Lim et al. , "Federated Learning in Mobile Edge Networks: A Comprehensive Survey," IEEE Communications Surveys & Tutorials , vol. 22, no. 3, p. 2031, Jan. 2020, doi: 10.1109/comst.2020.2986024.
[29] D. Kreković, P. Krivić, I. P. Žarko, M. Kušek, and D. Le-Phuoc, "Reducing Communication Overhead in the IoT-Edge-Cloud Continuum: A Survey on Protocols and Data Reduction Strategies," arXiv (Cornell University) , Apr. 2024, doi: 10.48550/arxiv.2404.19492.
[30] S. Puppala, I. Hossain, M. J. Alam, S. Talukder, Z. Talukder, and S. Bahauddin, "SCALE: Self-regulated Clustered federAted LEarning in a Homogeneous Environment," arXiv (Cornell University) , Jul. 2024, doi: 10.48550/arxiv.2407.18387.
[31] X. Yan et al. , "Sequential Federated Learning in Hierarchical Architecture on Non-IID Datasets," arXiv (Cornell University) , Aug. 2024, doi: 10.48550/arxiv.2408.09762.
[32] S. M. Azimi-Abarghouyi and V. Fodor, "Quantized Hierarchical Federated Learning: A Robust Approach to Statistical Heterogeneity," arXiv (Cornell University) , Mar. 2024, doi: 10.48550/arxiv.2403.01540.
[33] R. Zhagypar, N. Kouzayha, H. ElSawy, H. Dahrouj, and T. Y. Al-Naffouri, "UAV-assisted Unbiased Hierarchical Federated Learning: Performance and Convergence Analysis," arXiv (Cornell University) , Jul. 2024, doi: 10.48550/arxiv.2407.07739.
[34] W. Gao, O. Tavallaie, S.-H. Chen, and A. Y. Zomaya, "Federated Learning as a Service for Hierarchical Edge Networks with Heterogeneous Models," in Lecture notes in computer science , Springer Science Business Media, 2024, p. 85. doi: 10.1007/978-981-96-0805-8_6.
[35] R. Mirzaeifard and S. Werner, "Smoothing ADMM for Non-convex and Non-smooth Hierarchical Federated Learning," arXiv (Cornell University) , Mar. 2025, doi: 10.48550/arxiv.2503.08869.
[36] K. D. Cooper and M. R. Geller, "Advancing Personalized Federated Learning: Integrative Approaches with AI for Enhanced Privacy and Customization," arXiv (Cornell University) , Jan. 2025, doi: 10.48550/arxiv.2501.18174.
[37] A. Shrivastava, "Privacy-Centric AI: Navigating the Landscape with Federated Learning," International Journal for Research in

Applied Science and Engineering Technology , vol. 12, no. 5, p. 357, May 2024, doi: 10.22214/ijraset.2024.61000.
[38] J. Sen, H. Waghela, and S. Rakshit, "Privacy in Federated Learning," arXiv (Cornell University) , Aug. 2024, doi: 10.48550/arxiv.2408.08904.
[39] J. Li, Q. Sun, and F. Sun, "Enhancing Privacy-Preserving Intrusion Detection in Blockchain-Based Networks with Deep Learning," Data Science Journal , vol. 22, Jan. 2023, doi: 10.5334/dsj-2023-031.
[40] P. Hamedi, R. Razavi–Far, and E. Hallaji, "Federated Continual Learning: Concepts, Challenges, and Solutions," arXiv (Cornell University) , Feb. 2025, doi: 10.48550/arxiv.2502.07059.
[41] R. Rahman, "Federated Learning: A Survey on Privacy-Preserving Collaborative Intelligence," arXiv (Cornell University) , Apr. 2025, doi: 10.48550/arxiv.2504.17703.
[42] S. Bottoni, G. Zizzo, S. Braghin, and A. Trombetta, "Verifiability and Privacy in Federated Learning through Context-Hiding Multi-Key Homomorphic Authenticators," arXiv (Cornell University) , Sep. 2025, doi: 10.48550/arxiv.2509.05162.
[43] Y. Shanmugarasa, H.-Y. Paik, S. S. Kanhere, and L. Zhu, "A systematic review of federated learning from clients' perspective: challenges and solutions," Artificial Intelligence Review , vol. 56. Springer Science+Business Media, p. 1773, Aug. 07, 2023. doi: 10.1007/s10462-023-10563-8.
[44] D. Wasif, D. Chen, S. Madabushi, N. Alluru, T. J. Moore, and J.-H. Cho, "Empirical Analysis of Privacy-Fairness-Accuracy Trade-offs in Federated Learning: A Step Towards Responsible AI," arXiv (Cornell University) , Mar. 2025, doi: 10.48550/arxiv.2503.16233.
[45] Z. Ye, W. Luo, Q. Zhou, and Y. Tang, "High-Fidelity Gradient Inversion in Distributed Learning," in Proceedings of the AAAI Conference on Artificial Intelligence , Association for the Advancement of Artificial Intelligence, Mar. 2024, p. 19983. doi: 10.1609/aaai.v38i18.29975.
[46] E. Hosseini, S. Chen, and A. Khisti, "Secure Aggregation in Federated Learning using Multiparty Homomorphic Encryption," arXiv (Cornell University) , Mar. 2025, doi: 10.48550/arxiv.2503.00581.
[47] A. Wainakh, A. S. Guinea, T. Grube, and M. Mühlhäuser, "Enhancing Privacy via Hierarchical Federated Learning," p. 344, Sep. 2020, doi: 10.1109/eurospw51379.2020.00053.
[48] M. Sarhan, W. W. Lo, S. Layeghy, and M. Portmann, "HBFL: A hierarchical blockchain-based federated learning framework for collaborative IoT intrusion detection," Computers & Electrical Engineering , vol. 103, p. 108379, Sep. 2022, doi: 10.1016/j.compeleceng.2022.108379.
[49] S. A. Farooqi, A. A. Rahman, and A. Saad, "Advanced Privacy-Utility Optimization Techniques in Federated Learning with Differential Privacy for IoMT – A Review," International Journal of Interactive Mobile Technologies (iJIM) , vol. 19, no. 19. kassel university press, p. 134, Oct. 07, 2025. doi: 10.3991/ijim.v19i19.57619.
[50] Q. Cui et al. , "Overview of AI and Communication for 6G Network: Fundamentals, Challenges, and Future Research Opportunities," arXiv (Cornell University) , Dec. 2024, doi: 10.48550/arxiv.2412.14538.
[51] Y. Feng, Y. Qi, H. Li, X. Wang, and J. Tian, "Leveraging federated learning and edge computing for recommendation systems within cloud computing networks," p. 28, Jul. 2024, doi: 10.1117/12.3034773.
[52] S. Narkedimilli, A. Sriram, and S. Raghav, "FL-DABE-BC: A Privacy-Enhanced, Decentralized Authentication, and Secure Communication for Federated Learning Framework with Decentralized Attribute-Based Encryption and Blockchain for IoT Scenarios," arXiv (Cornell University) , Oct. 2024, doi: 10.48550/arxiv.2410.20259.
[53] R. Xu, S. Gao, C. Li, J. Joshi, and J. Li, "Dual Defense: Enhancing Privacy and Mitigating Poisoning Attacks in Federated Learning," arXiv (Cornell University) , Feb. 2025, doi: 10.48550/arxiv.2502.05547.
[54] Y. H. Pan, C. Zheng, W. He, J. Yang, H. Li, and W. Liming, "FedSHE: privacy preserving and efficient federated learning with adaptive segmented CKKS homomorphic encryption," Cybersecurity , vol. 7, no. 1, Jul. 2024, doi: 10.1186/s42400-024-00232-w.
[55] A. Madi, O. Stan, A. Mayoue, A. Grivet-Sebert, C. Gouy-Pailler, and R. Sirdey, "A Secure Federated Learning framework using Homomorphic Encryption and Verifiable Computing," p. 1, May 2021, doi: 10.1109/rdaaps48126.2021.9452005.
[56] D. Commey, S. Hounsinou, and G. V. Crosby, "Securing Health Data on the Blockchain: A Differential Privacy and Federated Learning Framework," arXiv (Cornell University) , May 2024, doi: 10.48550/arxiv.2405.11580.
[57] M. J. Mia and M. H. Amini, "QuanCrypt-FL: Quantized Homomorphic Encryption with Pruning for Secure Federated Learning," arXiv (Cornell University) , Nov. 2024, doi: 10.48550/arxiv.2411.05260.
[58] W. Li, K. Fan, J. Zhang, H. Li, W. Y. B. Lim, and Q. Yang, "Enhancing Security and Privacy in Federated Learning using Update Digests and Voting-Based Defense," arXiv (Cornell University) , May 2024, doi: 10.48550/arxiv.2405.18802.
[59] N. Nazemi et al. , "ACCESS-FL: Agile Communication and Computation for Efficient Secure Aggregation in Stable Federated Learning Networks," arXiv (Cornell University) , Sep. 2024, doi: 10.48550/arxiv.2409.01722.
[60] X. Yang, Z. Liu, X. Tang, R. Lu, and B. Liu, "An Efficient and Multi-private Key Secure Aggregation for Federated Learning," arXiv (Cornell University) , Jun. 2023, doi: 10.48550/arxiv.2306.08970.
[61] C. Jin et al. , "DMAFL: Effective defense against malicious attacker federated learning framework via blockchain and TFHE," Journal of King Saud University - Computer and Information Sciences , vol. 37, no. 8, Sep. 2025, doi: 10.1007/s44443-025-00256-3.
[62] L. Leite, Y. Santo, B. L. Dalmazo, and A. Riker, "Federated Learning under Attack: Improving Gradient Inversion for Batch of Images," arXiv (Cornell University) , Sep. 2024, doi: 10.48550/arxiv.2409.17767.
[63] J. Xia, Y. Zhang, Z. Yue, M. Hu, X. Wei, and M. Chen, "HierarchyFL: Heterogeneous Federated Learning via Hierarchical Self-Distillation," arXiv (Cornell University) , Dec. 2022, doi: 10.48550/arxiv.2212.02006.
[64] D. Shome, O. Waqar, and W. U. Khan, "Federated learning and next generation wireless communications: A survey on bidirectional relationship," Transactions on Emerging Telecommunications Technologies , vol. 33, no. 7, Feb. 2022, doi: 10.1002/ett.4458.
[65] A. Farajzadeh, A. Yadav, and H. Yanıkömeroğlu, "Multi-Tier Hierarchical Federated Learning-assisted NTN for Intelligent IoT Services," arXiv (Cornell University) , May 2023, doi: 10.48550/arxiv.2305.05463.